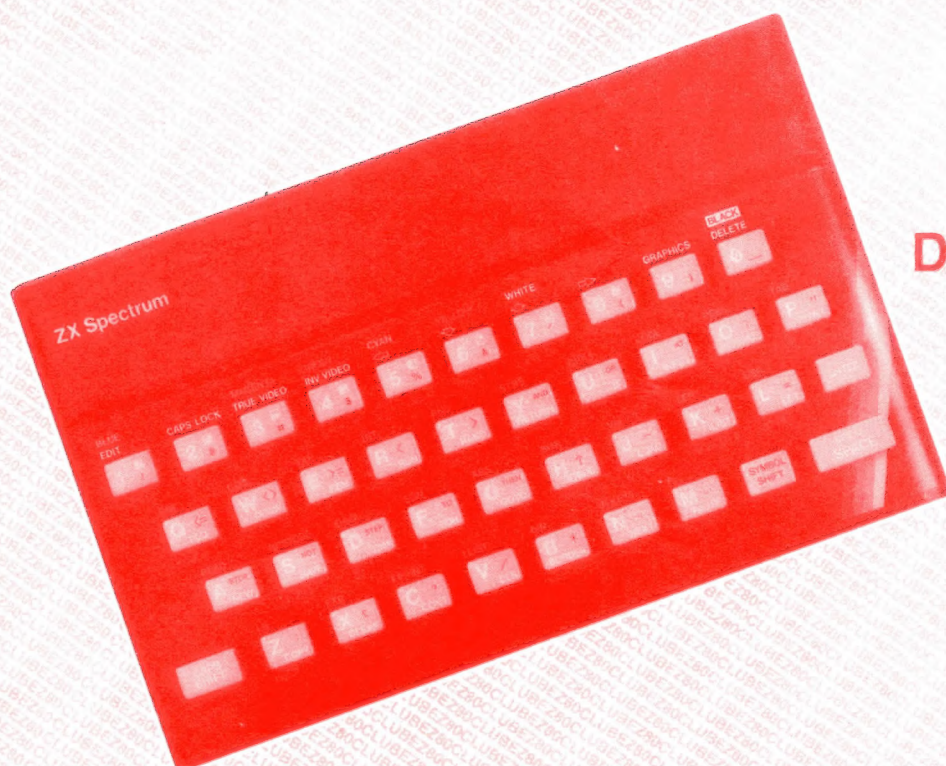


CLUBE

Z

80



Dezembro/83

N.º 15

NESTE NÚMERO

Editorial	1
Introdução à Linguagem Máquina (cont.)	2
Simulação (Parte III)	4
Enciclopédia da Linguagem Basic (cont.)	5

Programas ZX81/TMS 1000/Spectrum/Newbrain

Resolução de Triângulos	6
Calorias	7
Desenho	8
Ficheiro	9
Stars War	10
Traçado de Gráficos	12
Caracteres de Dupla Altura	13
Fire Fox (Correcções)	13
Vu-Meter (Correcções)	14
Jogo do Traço	15
Espaço Spectrum (Parte III)	15
Simulação de Circuitos Lógicos	19
Microdrive (cont.)	20

No Interior:

Cupão de Inscrição

Edição: Clube Z80

Fotocomposição: Fotomecânica Mabreu/Porto

Impressão: Gráfica Firmeza/Porto

Tiragem: 500 exemplares, Dezembro 1983

EDITORIAL

DEZEMBRO/83

Neste findar de 83 é tentador deitar uma olhadela ao que fizemos e ao que pretendemos fazer.

Em termos da n/ revista (ou jornal) passamos à impressão tipográfica e talvez tenhamos produzido um trabalho mais sistematizado. Tivemos a excepcional colaboração de Fernando Preces e pensamos ter conseguido agarrar a viragem do ZX 81 para o Spectrum.

É altura de perguntar: quantos possuidores do ZX 81 possuem agora o Spectrum? Venderam-se em Dezembro entre 5 000 a 10 000 Spectrum's. Se o n/ associado está registado como possuidor do ZX 81 e trocou de máquina durante este ano (posteriormente ao n/ inquérito) é importante nesta altura conhecer o grau de interesse que ainda existe para programas do ZX 81/TS 1 000.

O número de associados é de 204, ainda não suficiente para manter a impressão da revista mensal e suportar as despesas decorrentes do funcionamento dos serviços a prestar aos sócios (fotocópias, livros, programas, CTT's, etc.).

É importante fazer um esforço para aumentar o número de associados.

O número de jogos que existem no mercado, para usar com o Spectrum, atingiu já a centena. Gostaríamos que os n/ associados nos escrevessem, comentando os jogos e explicando o seu funcionamento, dado que, com o florescimento da «pirataria», os jogos chegam muitas vezes às mãos dos utilizadores sem instruções capazes.

Continuamos atentos às mensagens que nos chegam a solicitar programas para o **Apple** ou **New Brain** ou ainda versando problemas de Engenharia Civil ou outras áreas. Não se trata de esquecimento nosso — é uma questão de oportunidade. Quando isso for possível tentaremos sempre que a voz de cada um tenha sentido nas n/ páginas.

Finalmente, aqui vão as n/ desculpas pelo grande atraso com que este número vos chega às mãos.

Um grande abraço para todos
Alexandre Sousa/J. Magalhães
Isabel Cristina/Maria Irene

Rui Carvalho/Barreiro, colocou uma questão que nos parece importante para muitos utilizadores do SPECTRUM. (Atenção: Existe uma tendência dos constructores para abandonar o Basic) — **QUAIS AS ALTERNATIVAS PARA O BASIC? QUAIS AS VANTAGENS DO "PASCAL" OU "FORTH"?**

Neste momento existem **COMPILADORES** quer para a linguagem PASCAL, quer para **FORTH**, em cassette, para os possuidores do SPECTRUM.

FORTH oferece uma selecção de estruturas de controlo do programa, de modo a permitir que a execução seja condicional e interactiva.

Outro aspecto fundamental é que podem ser adicionadas novas especificações a uma "standard" já existente.

Esta linguagem tem já 10 anos de existência e começou por ser usada na resolução de problemas científicos críticos e em aplicações industriais. Em termos do entusiasta da programação, os aplausos vão para a eficiência com que a linguagem cumpre a sua função.

FORTH é simultaneamente:

- Linguagem de alto nível
- Linguagem "Assembly"
- Sistema Operatório
- Conjunto de ferramentas de desenvolvimento
- Filosofia de projectos de programação

Como LINGUAGEM, FORTH começa com um poderoso conjunto de comandos que proporcionam os mecanismos que permitem ao programador definir os seus próprios comandos.

A codificação, ao mais alto nível, parece uma descrição em língua inglesa.

É uma linguagem que pode ser usada para a resolução de problemas.

Exemplo:

```
MACHO?  
IF 1 HOMEM + !  
ELSE 1 MULHER + !  
THEN
```

Se o resultado de MACHO? é verdadeiro, a execução continua desde ELSE ou THEN, conforme o que estiver em primeiro lugar.

(Cont. no próximo número)

INTRODUÇÃO À LINGUAGEM MÁQUINA

ZX 81

FINAL DO 1.º CAPÍTULO — (Secção de Jogos) — Pequena recapitulação da matéria estudada

(Continuação)

JOGO 1 — (Rotina SCROLL *ESTE-OESTE*, e pequena rotina de *Pesquisa de colisão*)

```

1 REM (Reservar 28 caracteres)

8 GOSUB 1000
10 LET S = 0
20 LET A = CODE "<" (símbolo menor que)
30 LET B = A
50 LET Z = S
100 FOR D = S TO A * A
110 PRINT AT B, Z;
120 IF USR 16514 = A THEN GOTO 300
130 PRINT "O" (código 52)
140 IF INKEY$ = "F" THEN GOSUB 500
190 LET B = B - (INKEY$ = "7" AND B)
    + (INKEY$ = "6" AND B < A)
200 PRINT AT RND * A, 17; "<"
210 LET C = USR 16521
220 NEXT D
300 PRINT "■"; S; "PONTOS" (código gráf. 8)
310 FOR T = 0 TO 300
320 NEXT T
325 CLS
330 PRINT AT 12,0; "QUER CONTINUAR? (Diga S
    ou N)."
340 INPUT L$
345 CLS
350 IF L$ = "S" THEN GOTO 10
355 PRINT AT 12,0; "ATE À PRÓXIMA E... OBRIGADA."; W
500 FOR C = Z TO PI (símbolo  $\pi$ )
510 IF USR 16514 = A THEN LET S = S + 1
520 PRINT "-";
530 NEXT C
540 PRINT AT B, Z; "O" (Código 52 seguido de 4
    espaços)
550 RETURN
1000 PRINT AT 1,5; "BATALHA NO ESPAÇO"
1010 PRINT, "A SUA NAVE É ATACADA POR
    CENTE—"
1020 PRINT, "NAS DE PEQUENAS NAVES, CUJO
    FO—"
1030 PRINT, "GO É INOFENSIVO MAS, PERI-
    GOSAS"
1040 PRINT, "AO CHOQUE."
1050 PRINT, "A T 7 DESVIA A NAVE PARA CIMA
    E A"

```

```

1060 PRINT, "T 6 PARA BAIXO. A TF DISPARA."
1070 PRINT, "BOA SORTE..."
1080 PRINT, "PRIMA" "N/L" "PARA COMEÇAR."
1090 INPUT L$
1095 CLS
1099 RETURN
2000 SAVE "JOGO 1"
2010 RUN

```

NOTA: Este jogo pode correr numa máquina de 1 K RAM, se forem eliminadas as linhas (8, 310 a 360 e 1000 a 1099) acrescentando na linha 300 (: W).

Listagem do código máquina com início em 16514

(42, 14, 64, 78, 6, 0, 201, 1, 117, 23, 3, 42, 16, 64,
22, 0, 43, 126, 185, 40, 4, 114, 87, 24, 247, 16, 243, 201.)

Uma grande parte do programa escrito em Basic, incluindo a movimentação da nave e os seus disparos, a introdução de outros movimentos como acelerar ou travar, o aparecimento de outros obstáculos, etc., podem ser transformados em rotinas CM, que farão o jogo mais rápido e aliciante.

Em código máquina a movimentação por tecla é inquirida pela consulta cíclica a uma variável do sistema chamada Last K (última tecla premida) que depois é enviada por uma instrução CALL para a rotina de decodificação do teclado (na ROM), e o retorno entregue a uma rotina de movimento que soma ou subtrai o número 33 (salto de uma linha) à posição PRINT no ficheiro de projecção, ocupada pela nave. Quanto aos disparos, se a tecla F foi premida é comparada com o seu código por uma instrução CP, N e a informação conduzida com uma instrução JRZ ou CALL para a rotina de disparos. Não é difícil, e o leitor mais afoito ou mais experiente nestas andanças de programação em código máquina pode tentar transformar algumas dessas instruções do Basic. Quanto ao leitor que não possua ainda essa experiência, terá de ser paciente e aguardar o estudo dessas rotinas, para melhorar este e outros programas. A segunda rotina máquina deste jogo, que começa em 16521, é já nossa conhecida. Ela executa o Scroll Este-Oeste.

A primeira, com começo em 16514, é uma novidade de que ainda não tínhamos falado.

É muito pequena e apenas substitui 3 instruções do Basic e as respectivas operações aritméticas.

As linhas Basic 120 e 510, antes de ser introduzida a rotina máquina eram assim:

```

120 IF PEEK (PEEK 16398 + 256 * PEEK 16399) = A
    THEN GOTO 300

```


510 IF PEEK (PEEK 16398 + 256 * PEEK 16399) = A
THEN LET S = S + 1

Em termos de programa elas executam rigorosamente o mesmo trabalho só que a diferença em tempo de execução é de 3 para 1.

Interpretação da rotina em CM:

42	LD, HL, (NN)	} carrega HL com o endereço da posição PRINT no ficheiro de projecção
14		
64		
78	LD C, (HL)	carrega C com o conteúdo de HL
6	LD B, N	} coloca B a 0 para o retorno
0	0	
201	Ret	retorno ao Basic

Para uma melhor compreensão vamos interpretar o que está programado nas linhas Basic 120 e 510.

Linha 120

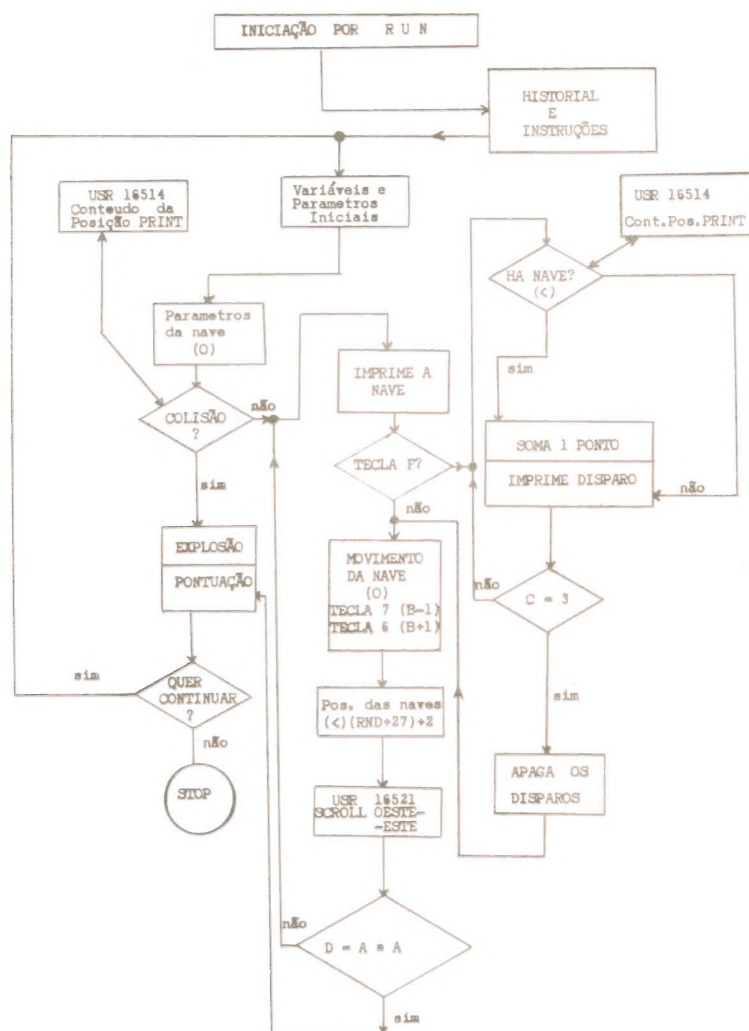
Se uma pequena nave (<) for ocupar o espaço da nave (O), temos uma colisão, e com GOTO 300, a explosão e a pontuação até ao momento, terminando o jogo. Por outras palavras: se o conteúdo de HL for igual ao código 19 (<) então há colisão.

Linha 510

Ao ser premida a tecla F, se uma pequena nave for ocupar o espaço destinado ao disparo (-), *nova posição PRINT do ficheiro*. A nave (<) é eliminada e ganha-se o ponto, com LET S = S + 1.

Mais para a frente, este jogo sofrerá as transformações compatíveis ao desenvolvimento do estudo da linguagem máquina efectuado, até atingir algumas das potencialidades de um jogo como o GROUND ATTACK.

Para habituar o leitor ao estudo e montagem dum programa, sobre o seu organigrama, um dos assuntos a ser tratado no próximo capítulo, acrescenta-se aqui o diagrama bloco deste jogo.



(Cont. no próximo número)

SIMULAÇÃO

PARTE III/DEZ. 83

Referimos anteriormente que a nossa próxima etapa é colocar o relógio no início do dia.

A linha de espera dos clientes será indicada pela variável L e será inicializada com o valor zero. A variável representativa das perdas de caixa será PC e o total de vendas para o dia será indicado pela variável TV; ambas serão colocadas em zero.

```
300 LET HT = 7 : LET MT = 0
```

```
310 LET L = 0 : LET PC = 0 : LET TV = 0
```

No início de cada hora, o programa designa o número de clientes. Para a hora "J", representará a chegada do cliente A(J). A cada cliente será atribuído um tempo de chegada em minutos (após a hora) e o computador irá escolher o tempo de chegada, usando o gerador de números aleatórios (RANDOM).

Na ausência de qualquer outra informação, assumimos que o cliente vai sair aleatoriamente uns minutos após a hora. Vamos ver como isto se processa em termos do computador:

No início de cada hora simulada, lançamos uma tabela, D(T), com 15 entradas, uma por cada período de 4 minutos (dentro de cada hora).

Esta tabela ou "array" deverá indicar quantos clientes chegam em cada intervalo de 4 minutos. Por exemplo, se $D(10) = 4$, então quatro clientes chegam entre o minuto 36 e o minuto 40 (dessa hora), ou seja, no décimo intervalo do tempo dessa hora.

O programa vai situar aleatoriamente cada um dos clientes A(J), num intervalo de 4 minutos, usando o gerador de números aleatórios. O nosso programa deverá testar o tempo para o início da hora. Isto será feito pela determinação de MT igual a zero (linha 410). No caso disso suceder, irá para a linha 1200, onde situa o cliente que chegou na hora exacta do minuto zero.

```
11 DIM D(15)
```

```
.....
410 IF MT = 0 THEN GOSUB 1200
```

```
.....
1200 FOR S = 1 TO 15
```

```
1220 NEXT S
```

```
1230 FOR I = 1 TO A(HT)
```

```
1240 LET X = INT (15 * RND) + 1
```

```
1250 LET D(X)=D(X)+1
```

```
1260 NEXT I
```

```
1270 RETURN
```

O programa irá agora progredir ao longo da hora simulada em segmentos de 4 minutos.

Para o segmento de ordem T, ele origina que o cliente D(T) chega ao bar.

Vamos assumir que metade destes clientes irão ficar e outra metade entra e sai logo.

O computador irá dar uma olhadela a cada um destes clientes e decidirá quais os que ficam e quais os que irão sair. Se o cliente vai ficar, será adicionado à linha do cliente que permaneceu no bar. Se o cliente entra e sai, o computador adiciona à variável PC o valor 57\$50, que significa um valor de caixa perdido. Quando o cliente permanece na linha, para ser atendido, será acrescentado o valor 57\$50 à variável TV (total de vendas). Finalmente, o tempo será actualizado e o mesmo procedimento será repetido no próximo segmento de 4 minutos.

Posto isto, teremos a finalidade desta simulação — analisar se este negócio exigirá ou não a admissão de outra pessoa, ou mesmo se deveremos mudar de negócio.

Vamos então ver a parte do programa onde é executada esta análise.

A linha 720 vai fazer o teste do tempo para o final do dia ($HT = 19$).

Quando isto se verifica, o programa vai para a linha 1500, onde o final do dia estatístico vai ser remetido para o ecran.

```
420 LET T = MT/4 + 1
```

```
430 FOR J = 1 TO D(T)
```

```
440 LET C = INT (2 * RND) + 1
```

```
450 IF C = 1 THEN GOTO 500
```

```
455 GOTO 600
```

```
490 REM 500-560 SAÍDA CLIENTE
```

```
500 IF RND > D(L) THEN GOTO 550
```

```
510 LET PC = PC + 57.50
```

```
520 GOTO 690
```

```
550 LET L = L + 1 : REM CLIENTE SAI
```

```
560 GOTO 690
```

```
590 REM 600-660 CLIENTE FICA
```

```
600 IF RND > P(L) THEN GOTO 640
```

```
610 LET PC = PC + 57.50 (CLIENTE PERDIDO)
```

```
620 GOTO 690
```

```
640 LET L = L + 1
```

```
690 NEXT J
```

```
700 IF L = 0 THEN GOTO 710
```

```
702 LET L = L - 1
```

```
704 LET TV = TV + 57.50 (CLIENTE GANHO)
```

```
710 GOSUB 1000 (ACTUALIZAR O TEMPO)
```

```
720 IF HT = 19 THEN GOTO 1500
```

```
725 GOTO 800
```

```
730 REM HT = 19 FINAL DO DIA
```

```
800 GOTO 410 (PRÓXIMO SEGMENTO DE
4 MINUTOS)
```



```

1500 PRINT "FINAL DO DIA ESTATÍSTICO"
1510 PRINT "VALOR DE VENDAS PERDIDO",
      PC + PC * 57.50
1520 PRINT "VALOR REALIZADO EM CAIXA", TV
1530 PRINT "LINHA FECHADA AO CLIENTE N.º", L
2000 END

```

O nosso programa simula a actividade de um simples dia. Para termos uma média estatística, em relação a um certo número de dias, teremos de repetir o programa.

Vamos escolher arbitrariamente 10 dias.

A variável D1 vai registar esse número de dias. A variável TP regista o valor de negócio perdido e a variável T6 representa o total ganho. Estas duas variáveis serão actualizadas no final de cada dia.

O "dia número ..." será representado por E e a mudança do dia será controlada pelo ciclo das linhas 290 e 1700:

```

      290 FOR E=1 TO 10
      1700 NEXT E

```

Estatisticamente, iremos calcular a média das perdas PC, o total de vendas/dias e o fecho da linha diária L. Teremos em conta os valores destas variáveis para todos os dias; estão representadas por L1, C1 e CL, respectivamente.

```

1500 LET L1 = PC + L1 + L * 57.50
1510 LET C1 = TV + C1
1520 LET CL = L + CL

```

As linhas 1800 a 1850 calculam as médias de L1, C1 e CL e mostram os seus resultados.

(Cont. no próximo número)

ENCICLOPÉDIA DA LINGUAGEM BASIC

ASC

A função ASC converte um character ou variável alfanumérica (string) no seu valor numérico decimal (código ASCII).

Por exemplo:

Print ASC ("A") apresentará o valor 65 no ecran, dado que esse é o seu valor em código ASCII.

Se pedir: Print ASC (A\$), obterá o código do primeiro character da variável A\$.

PROGRAMA 1

```

10 PRINT "O CÓDIGO ASCII PARA A LETRA
   'A' SERÁ = ";
20 PRINT ASC ("A")
30 IF ASC ("A") = 65 THEN 70
40 PRINT "TESTE FALHADO POR 'ASC'"
60 GOTO 99
70 PRINT "TESTE OK! PARA 'ASC'"
99 END

```

PROGRAMA 2

```

10 PRINT "ENTRADA PARA QUALQUER NÚ-
   MERO, LETRA OU CHARACTER"
20 INPUT A$
30 PRINT "O CÓDIGO ASCII PARA ";A$;"
   E = ";ASC(A$)
40 GOTO 20

```

ALGUNS COMPUTADORES QUE INCORPORAM A FUNÇÃO 'ASC' PODEM ACEITAR STRINGS SUPERIORES A UM CHARACTER, MAS SOMENTE O PRIMEIRO CHARACTER SERÁ CONVERTIDO NO RESPECTIVO CÓDIGO.

OBSERVAÇÃO

NAS MÁQUINAS SINCLAIR, ESTA FUNÇÃO TEM O NOME DE ... CODE ... PELO QUE PODE TESTAR O TEXTO DESTE MÊS, USANDO A FUNÇÃO 'CODE' EM VEZ DE 'ASC'.

(Cont. no próximo número)

RESOLUÇÃO DE TRIÂNGULOS

SPECTRUM/ZX81/TS 1000

Adapt. ALEXANDRE SOUSA
Porto



PROGRAMA PARA RESOLUÇÃO DE TRIÂNGULOS
PODE SER ADAPTADO SEM DIFICULDADE A QUALQUER MÁQUINA
FOI ESCRITO PARA O SINCLAIR SP
ECTRUM

PARA OS POSSUIDORES DO ZX81
CHAMA-SE A ATENÇÃO P/ AS LINHAS
TIPO...INPUT'LADO='

PODEM SER SUBSTITUIDAS POR:

```
PRINT'LADO='
INPUT L(1)
```

TAMBEM DEVEM NOTAR QUE O SIM-
BOLO * DEVE SER SUBSTITUIDO
POR **

ESTE PROGRAMA FOI ADAPTADO DE
OUTRO EXISTENTE P/APPLE II

EXEMPLO PROBLEMA TIPO 2

LADO=3.75 ÂNGULO=80

LADO=3.75

RESULTADOS

```
LADO 1=13.283
ÂNGULO OPÓSTO=80 GRAUS
LADO 2=8.700
ÂNGULO OPÓSTO=45 GRAUS
LADO 3=8.700
ÂNGULO OPÓSTO=45 GRAUS
```

EXEMPLO PROBLEMA TIPO 3

LADO=10 LADO=13 LADO=14

RESULTADOS

```
LADO 1=10
ÂNGULO OPÓSTO=43.278 GRAUS
LADO 2=13
ÂNGULO OPÓSTO=83.087 GRAUS
LADO 3=14
ÂNGULO OPÓSTO=73.883 GRAUS
```

```
1 REM Programa Educacional
2 REM p/construir um triangulo
3 REM das as partes conheci
das
4 REM o programa calcula as p
rtes desconhecidas
1000 CLS
1100 DIM A(3)
1200 DIM L(3)
1300 REM A=ÂNGULO L=LADO
1400 PRINT AT 10,0;"1- P L B"
1400 PRINT AT 12,0;"2- P L B"
1400 PRINT AT 14,0;"3- P L B"
1400 PRINT AT 16,0;"4- P L B"
1400 PRINT AT 18,0;"5- L L L"
1500 PRINT AT 20,10;"ESCOLHA...?"
```

```
160 INPUT Y#
1700 IF Y#="1" THEN GO TO 1000
1800 IF Y#="2" THEN GO TO 1000
1900 IF Y#="3" THEN GO TO 2000
2000 IF Y#="4" THEN GO TO 2000
2100 IF Y#="5" THEN GO TO 3000
2200 GO TO 100
2300 REM 3
2400 CLS
2500 INPUT "LADO=";L(3)
2600 INPUT "ÂNGULO/GRAUS=";A(1)
2700 GO SUB 2000
2800 LET A(1)=A
2900 INPUT "LADO=";L(2)
3000 LET L(1)=SQR (L(3)^2+L(2)^2-
3100 *L(3)*L(2)*COS (A(1)))
3200 LET A(2)=SIN (A(1))/L(1)*L
3300 LET A(2)=ATN (A(2)/SQR (-A
3400 *A(2)+1))
3500 LET A(3)=PI-A(1)-A(2)
3600 GO TO 440
3700 REM 3
3800 CLS
3900 INPUT "ÂNGULO=";A
4000 GO SUB 2000
4100 LET A(3)=A
4200 INPUT "ÂNGULO=";A
4300 GO SUB 2000
4400 LET A(2)=A
4500 INPUT "LADO=";L(3)
4600 LET A(1)=PI-A(2)-A(3)
4700 GO TO 2000
4800 REM 4
4900 CLS
5000 INPUT "ÂNGULO=";A(1)
5100 INPUT "LADO=";L(2)
5200 INPUT "ÂNGULO=";A(2)
5300 LET T=L(2)*SIN (A(1))
5400 IF L(1)/T THEN GO TO 300
5500 LET L(3)=SQR (L(2)^2+T^2-
5600 *L(2)*T)
5700 IF L(1)/T THEN GO TO 300
5800 LET L(3)=L(1)+T^2-
5900 *L(3)+Y
6000 GO TO 220
6100 REM 5
6200 CLS
6300 INPUT "ÂNGULO=";L(1)
6400 INPUT "LADO=";L(2)
6500 INPUT "LADO=";L(3)
6600 LET A(1)=(L(2)^2+L(3)^2-L(1)^2)/
6700 *2/L(2)/L(3)
6800 LET A(1)=ATN (A(1)/SQR (1-A
6900 (1)+A(1)+1))+1.5708
7000 GO TO 220
7100 CLS
```



```

442 PRINT "RESULTADOS"
443 FOR I=1 TO 3
444 IF A(I)<0 THEN GO TO 520
445 LET A(I)=A(I)*57.29573
446 PRINT "LADO "I"="INT (L*
I)+1000+.5)/1000
448 PRINT "ANGULO OPOSTO="INT
(I*(I+1000+.5)/1000)" GRAUS"
450 NEXT I
452 PRINT AT 20,0,"0.TECLA F/NO
VO PROBLEMA"
470 INPUT Y$
480 GO TO 100
500 I=5
502 PRINT "PROBLEMA SEM SOLUCAO"

524 PAUSE 100
530 GO TO 100

```

```

1000 REM 1
1001 CLS
1010 INPUT "1)ANGULO/GRAUS=";A
1020 GO SUB 2000
1030 LET A(1)=A
1037 INPUT "LADO=";L(3)
1038 INPUT "2)ANGULO/GRAUS=";A
1034 GO SUB 2000
1035 LET A(2)=A
1040 LET A(3)=PI-A(1)-A(2)
1050 LET L(1)=L(3)*SIN (A(1))/SIN
N (A(3))
1055 LET L(2)=L(3)*SIN (A(2))/SIN
N (A(3))
1060 GO TO 440
2000 REM FACTOR CONU.GR--RAD
2010 LET R=A*.0174533
2020 RETURN

```

CALORIAS

SPECTRUM/ZX 81/TMS 1000

CARLOS SILVA

Tomar

Este programa diz-nos o total de calorias/dia necessárias a um indivíduo, mediante a idade, sexo, peso, altura, horas de sono, horas de repouso e horas de trabalho diário.

```

1000 DE=35
1001 X= (M/F)...M
1002 P=...KG=80
1003 ALTURA=...CM=164
1004 SONO=7
1005 DEPO=3
1006 LITROS=1
1007 TRABALHO=10
1008 TRABALHO=0

```

TOTAL Kcal/dia = 3388,4848

```

1000 CARLOS SILVA+TOMAR+
1001 DE=35
1002 P=...KG=80
1003 ALTURA=...CM=164
1004 SONO=7
1005 DEPO=3
1006 LITROS=1
1007 TRABALHO=10
1008 TRABALHO=0
1009 TOTAL Kcal/dia = 3388,4848
1010 INPUT "IDADE="
1011 INPUT "SEXO (M/F)..."
1012 INPUT "PESO...KG="

```

```

600 INPUT I
601 PRINT I
602 PRINT "ALTURA (CM)="
603 INPUT U
604 PRINT U
605 IF S$="F" THEN GO TO 55
606 LET M=6*U+U*U/D/K*F
607 GO TO 600
608 LET M=6*U+U*U/D/K*H
609 PRINT "HORAS DE SONO="
610 INPUT O
611 PRINT O
612 PRINT "HORAS DE REPOUSO="
613 INPUT D
614 PRINT D
615 PRINT "HORAS DE TRABALHO LI
616 D=0"
617 INPUT E
618 PRINT E
619 PRINT "HORAS DE TRABALHO ME
620 D=D+1"
621 INPUT F
622 PRINT F
623 PRINT "HORAS DE TRABALHO FE
624 D=D+1"
625 INPUT G
626 PRINT G
627 IF 24=D+O+E+F+G THEN GO TO
630
630 PRINT "ERRO DE CALCULO...30
MAR DE NOVO AS HORAS"
670 INPUT S$
680 CLS
690 GO TO 55
700 PRINT AT 20,0;"TOTAL Kcal/dia
1010 INT C=M+D+E+F+G+4*G/41+
1.184

```

ESTOU INTERESSADO EM TROCAR PROGRAMAS E IDEIAS... POSSUO CERCA DE 50 PROGRAMAS PARA O ZX SPECTRUM, DESDE UTILITÁRIOS (ASSEMBLER, COMPILADOR FORTH, ETC.), ATÉ JOGOS COM EFEITOS GRÁFICOS E SONOROS (MINEIRO MANÍACO, AQUA-PLANE, ETC.), E JOGOS DE ESTRATÉGIA (BATTLE OF BRITAIN, ESPIONAGE ISLAND, ETC.). TROCO-OS POR PROGRAMAS QUE AINDA NÃO POSSUA, COMO POR EXEMPLO O SPEAKEASY, COMPILADOR BASIC E OUTROS.

ANTÓNIO JOÃO GOMES NUNES
Rua do Til, 72
9000 FUNCHAL

VENDO ZX 81, EXTENSÃO DE MEMÓRIA DE 32 K PARA MOTHERBOARD, MOTHERBOARD, CIRCUITO GERADOR DE SONS, FONTE DE ALIMENTAÇÃO PARA CIRC. GERADOR DE SONS, 8 CASSETES GRAVADAS COM PROGR. DE JOGOS E OUTROS (NO VALOR DE 5 000\$00), LITERATURA PARA ZX 81. TUDO POR ESC.: 16 000\$00.

CARLOS SILVA
Bairro da Caixa, Lote 5-r/c Esq.
2300 TOMAR

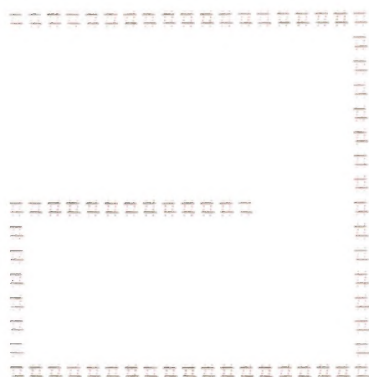
DESENHO

SPECTRUM/ZX 81/TMS 1000

CARLOS SILVA

Tomar

Este programa desenha um caracter que pode ser movimentado para cima ou baixo e para a esquerda ou direita. Para mudar de caracter, prima 0 (zero).



```

      CARLOS SILVA* TOMAR+
      DESENHO"
      A,B,A#
      THINKY#
      THEN GO TO 50
      THEN LEFT TO 50
      THEN LEFT TO 50
      THEN GO TO 30
  
```

VENDO ZX 81 COM UNIDADE 16 K,
TRANSFORMADOR MANUAL E CABOS
POR ESC.: 10 000\$00
(MOTIVO FALTA DE USO)

JORGE SANTOS MONTEIRO

Rua da Fonte — Vilarinho/Cacia
3 8 0 0 A V E I R O

ESTE DESENHO FOI FEITO COM MUITA
IMPLICIDADE USANDO ESTE PROGRAMA

NOVOS PROGRAMAS

SPECTRUM

- **COOKIE** — Os ingredientes que o cozinheiro utiliza para fazer o seu bolo recusam-se a entrar na tigela e agriem o "cookie". Este atrai-lhes com bombas de farinha. Mas há ainda o "Monstro do lixo" a atacá-lo.
- **SPACE WAR** — Simulação de uma batalha espacial.
- **ZOOM** — Você vai repelir ataques aéreos, terrestres e marítimos.
- **PSSST** — Usando insecticidas, o jardineiro tenta proteger a sua planta de ser atacada por pragas esfomeadas.
- **TRANZAM** — Você controla um bólido que atravessa o continente americano. Mas outros carros perseguem-no.
- **INVASÃO GALÁCTICA** — Para deter a invasão você controla 15 mísseis.
- **DROP A BRICK** — Destrua uma parede de tijolos com uma bola.
- **SILHOUETTE** — Desenhe qualquer figura e use a impressora para guardar uma cópia.
- **ARMAGEDON** — Destrua os satélites inimigos e evite a destruição das cidades, repelindo os mísseis nucleares.
- **FRENZIE** — Você está num labirinto e se tocar nas paredes carregadas com 20 000 000 volts, pode ficar destruído — fuja dos robots assassinos! Para isso possui um "Laser" que dispara no sentido em que se desloca.
- **VIOLENT UNIVERSE** — Guerra no espaço.
- **ZIP ZAP** — Um robot vai explorar a possível colonização do universo. A sua energia provém de células de fuel e a defesa é feita com laser. Pode usar um escudo protector, activado por 10 unidades de energia. Quando passa de um sector para outro, tem um bónus de energia. O "Score" depende do sector e do número de inimigos abatidos.

Preço-programa 400\$00

FICHEIRO

SPECTRUM

Autor: RICARDO JORGE VAZ
Porto

O programa FICHEIRO tem capacidade de 96 discos. Depois de ter introduzido o programa no computador, aparecerá um MENU no ecrã com 5 hipóteses:

- 1.^a — criar uma ficha;
- 2.^a — consultar (todo o ficheiro ou só uma ficha);
- 3.^a — alterar;
- 4.^a — verificar e gravar;
- 5.^a — imprimir.

Nota: Com ligeiras alterações, este programa pode adaptar-se ao ZX81 ou TMS1000.

NOVO LIVRO

LA CONDUITE DU ZX SPECTRUM

HARTNELL Tim e JONES Dilwyn
Eyrolles, Paris, 1983

— O teclado; Instruções (PRINT, TAB, SAVE, VERIFY, MERGE...); Variáveis; Cor e Gráficos; N.ºs Cadeias de caracteres; PEEK e POKE...

Preço (Fotocópias) 860\$00

Desconto de 10% para Sócios do Clube

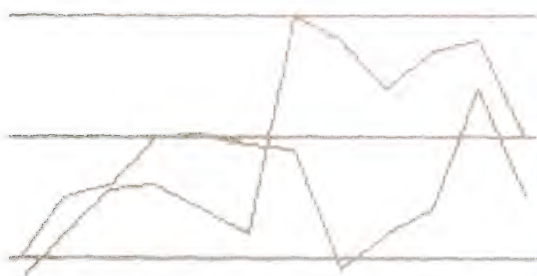
TRAÇADO DE GRÁFICOS

SPECTRUM

```

10 LET s=PI+50
15 DIM y(12)
20 INPUT INVERSE 1;"Nome do grafico "; INVERSE 0;"(32 caracteres) "; LINE t0:
IF LEN t0>32 THEN GO TO 10
30 PRINT AT 0,0;t0; FOR m=LEN t0 TO 31: PRINT " "; NEXT m
40 FOR n=0 TO 22: PRINT AT 20,n;" "; NEXT n: PRINT
50 LET a$="JFMAMJJASOND"
60 FOR n=1 TO 12: PRINT a$(n);" "; NEXT n
70 PRINT AT 2,24; INVERSE 1;"Entrada";AT 3,24; INVERSE 0;"max.150"
80 FOR n=1 TO 12: PRINT AT n+4,24;a$(n);" = ";y(n);" "; NEXT n
100 FOR n=1 TO 12
110 INPUT INVERSE 1;"Entradas"; INVERSE 0;" (s=fim) ";y(n); IF y(n)=s THEN GO
TO 170
120 IF y(n)>150 OR y(n)<0 THEN PRINT AT 18,24; FLASH 1;" Fora ";AT 19,24;" do
";AT 20,24;" limite"; PAUSE 100; FOR m=18 TO 20: PRINT AT m,24; FLASH 0;"
"; NEXT m: GO TO 110
130 PRINT AT n+4,24; INVERSE 1;a$(n); INVERSE 0;AT n+4,28;y(n)
140 IF n=1 THEN PLOT 4,y(1)+15
150 IF n>1 THEN DRAW 16*(n-1)-16*(n-2),y(n)-y(n-1)
160 NEXT n
170 INPUT "Traçado da escala ? "; LINE g$. IF g$<>"s" THEN GO TO 200
180 FOR y=0 TO 100 STEP 50: PLOT 0,65+y; DRAW 184,0; NEXT y
200 INPUT "Copiar na impressora ? "; LINE p$
210 IF p$="s" THEN COPY
300 INPUT "novos dados ? "; LINE n$
310 IF n$<>"s" THEN STOP
320 INPUT "Grafico sobreposto ? "; LINE s$
330 IF s$="s" THEN GO TO 10
340 CLS : GO TO 10

```



Entrada	
max. 150	
100	400
90	300
80	200
70	100
60	0
50	100
40	200
30	300
20	400
10	500
0	600

J F M A M J J A S O N D

TRAÇADO DE ESCALAS (ANUAL)

Este programa elabora escalas do tipo comercial com a vantagem de, com maior facilidade, verificar a variação de valores.

Assim terá apenas de introduzir os valores correspondentes aos meses do ano representados pelas iniciais. Em seguida, ser-lhe-á dada a opção do traçado na escala e, se deseja, cópia do gráfico.

Por fim, poderá dar entrada de novos dados, conseguindo obter a sobreposição dos gráficos caso o pretenda.

PREÇO DOS N.º ANTERIORES DOS JORNAIS CLUBE Z 80:

N.º 0 ATÉ 12 *	100\$00
N.º 13 E SEGUINTE	150\$00

(Preços sujeitos a alteração)

Podem ser solicitados ao CLUBE Z 80 para envio à cobrança ou adquiridos directamente na sede do CLUBE

* Venda Exclusiva a Sócios

CARACTERES DE DUPLA ALTURA

SPECTRUM

Adapt. ALEXANDRE SOUSA

Porto

Descritivo: O método que o autor (Kevin Ball's) escolheu, está baseado no estabelecimento de uma rotina, que examina a área do 'Display File' e expande os bytes de modo a produzir caracteres de dupla altura.

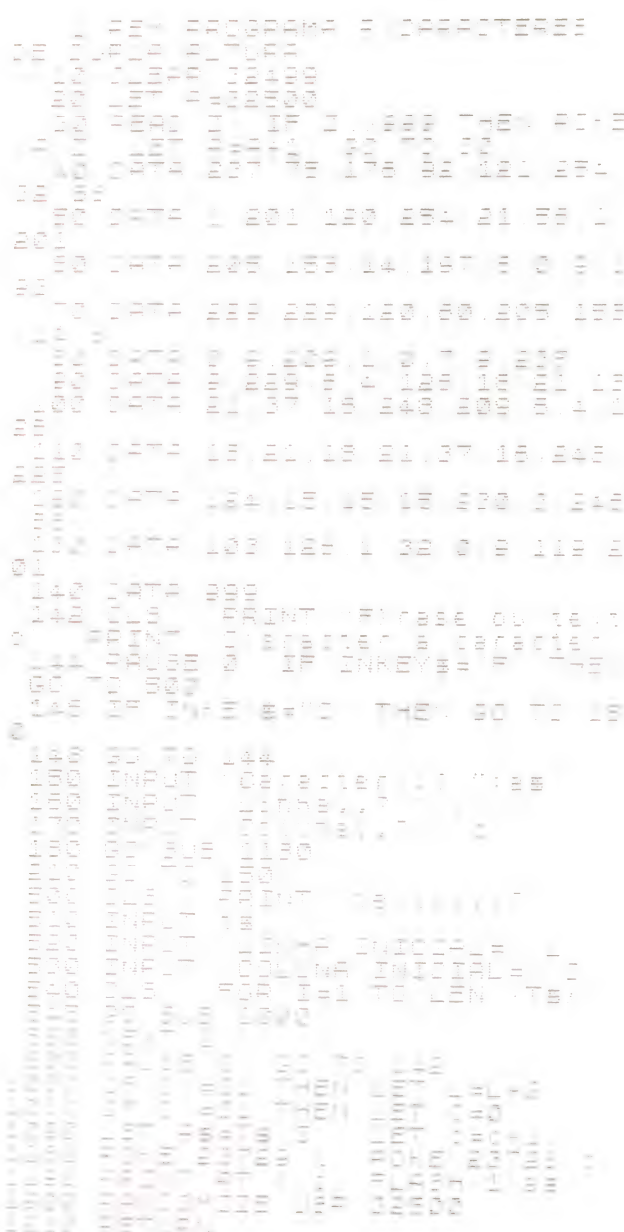
OBSERVAÇÃO: TAMBÉM POSSUÍMOS IDÊNTICO PROGRAMA PARA O ZX81/TS 1000, E PODEMOS ENVIAR UMA CÓPIA AOS ASSINANTES QUE O PEDIREM.

MÉTODO: Inicialmente escrevemos o carácter ou o texto no ecrã. Indicamos a posição da linha e a coluna onde queremos o carácter (ou o primeiro carácter do texto). Em seguida, a rotina é chamada na linha 1200... **RANDOMIZE USR 32500.**

Observe que cada carácter ocupa duas linhas. O programa está projectado de forma a que o utilizador não se preocupe com a posição da Memória RAM em que vai alojar a rotina, quer tenha 48 K ou 16 K.

SE PEDIR PARA ESCREVER NA LINHA 21, OS CARACTERES NÃO PODEM SER EXPANDIDOS POR FALTA DE ESPAÇO NO ECRAN.

COMO O CÓDIGO DECIMAL ESTÁ GUARDADO NAS LINHAS DE 'DATA', PODERÁ USAR ESTE PROGRAMA DENTRO DOS PRÓPRIOS PROGRAMAS, NECESSITANDO APENAS DE RENUMERAR AS LINHAS E CHAMAR ESTA ROTINA.



FIRE FOX (CORRECÇÃO)

(V. n.º 12, pág. 15)

A listagem publicada no n.º 12 está incorrecta. Aqui vai a listagem devidamente corrigida.

```

1 PAPER 0: INK 7: CLS : BORDE
2 CLEAR 55000: LET ss=0: LOAD
  CODE 65100
3 GO TO 2000
4 LET s=0

```

```

5 LET x=0: LET y=0
6 LET b=0: LET by=0: LET bx=0
10 FOR n=16 TO 21: PRINT AT n,
  0: PAPER 5: INK 5: "": NEXT n
20 LET f=0: LET x=10: LET y=7
30 PRINT AT 21,11: INK 0: PAPER
  5: INVERSE 1: "FIREFOX": AT 19,11
  1: INVERSE 0: "score ";s: AT 17,11
  FLASH 1: "Hi score "; FLASH 0:
  INVERSE 1: s:
100 FOR z=1 TO 200

```


JOGO DO TRAÇO

NEW BRAIN

Adapt. do BEGINNER'S GUIDE

Por Isabel Cristina

PORTO

```

10 KLM ***Jogo do traço***
20 OPEN#0,"85": PUT 23,65: CLOSE#1
30 OPEN#1,11,"n130"
40 CLEAR: x3=168
50 plot colour(1)
60 plot range(200,100),centre(12,30)
70 plot background(1),wipe
80 PUT 31: REM limpar o ecran
90 PRINT "Jogo do TRAÇO "
100 PRINT PRINT "Vamos jogar por turnos". " - Tu comesas"
110 PRINT "De cada vez só podes escolher"
120 PRINT "1 - 2 ou 3 traços do quadro"
130 PRINT "O que ficar com o ultimo "
140 PRINT "Perde."
150 GOSUB 1000: x2=0
160 PRINT PRINT "Quantos traços escolhes ?"
170 PRINT "Introduz o numero e acciona NEW LINE "
180 INPUT "": s$
190 PUT 31
200 IF s$="1" THEN x3=x2
210 IF s$="2" THEN x3=x2+14
220 IF s$="3" THEN x3=x2+28
230 IF s$="1" OR s$="3" THEN GOTO 160
240 GOSUB 1200
250 plot colour(2): GOSUB 1000
256 REM
257 RANDOMIZE
258 R = INT(RND * 4)
259 IF R = 0 THEN GOTO 258
260 IF x2 = 126 THEN GOTO 430
261 IF x2 = 140 THEN GOTO 440
262 IF x2 = 154 THEN GOTO 450
263 IF x2 = 168 THEN GOTO 630
264 IF x2 >= 182 THEN GOTO 680
270 IF R = 1 THEN GOTO 300
280 IF R = 2 THEN GOTO 310
290 IF R = 3 THEN GOTO 320
300 x3=x2+28 a=3: GOTO 330
310 x3=x2+14 a=2: GOTO 330
320 x3=x2 a=1
330 PRINT "E a minha vez " PRINT "Escolho "
340 GOSUB 1200: GOSUB 1000
350 PUT 31

```

```

360 PRINT "Joga outra vez " PRINT
370 PRINT "Quantos traços escolhes"
380 PRINT "Introduz o numero e acciona NEW LINE"
390 GOTO 180
410 IF s$="2" THEN GOTO 440
420 IF s$="3" THEN GOTO 450
430 x3=x2+28 a=3: GOTO 460
440 x3=x2+14 a=2: GOTO 460
450 x3=x2 a=1
460 PUT 31
470 PRINT "Sou eu - e escolho " a
480 GOSUB 1200: GOSUB 1000: PRINT PRINT
490 PRINT "Ficou o ultimo para ti"
500 PRINT PRINT "GANHEI!"
510 PRINT "Queres jogar outra vez ?"; "(S,N)"
520 INPUT "": s$
540 IF s$="S" OR s$="s" THEN GOTO 40
550 IF s$="N" OR s$="n" THEN GOTO 580
560 GOTO 530
580 PUT 31: plot wipe: PRINT
590 PRINT "ADEUS "
600 CLOSE#1
610 END
630 PUT 31: PRINT PRINT "Fica o ultimo"
640 PRINT "Para mim." PRINT
650 PRINT "OH! - GANHASTE"
660 PRINT GOTO 510
680 PUT 41: PRINT
690 PRINT "Escolheste muito"
700 PRINT PRINT "GANHEI"
710 PRINT GOTO 510
1000 REM
1001 FOR i = x2 TO x3 STEP 14
1010 plot place(x2)
1020 plot move(x2,50)
1030 plot move(x2+4,50)
1040 plot move(x2+4,0)
1050 plot move(x2,0)
1060 FOR xi=x2 TO x3
1070 plot place(x2+1),move(x2+4,x2+1)
1080 NEXT xi
1090 NEXT i
1100 x2 = x3 + 14
1110 RETURN
1200 REM delay loop
1210 FOR z = 1 TO 500
1220 NEXT z
1230 RETURN

```

ESPAÇO SPECTRUM

Autor: FERNANDO D'ALMEIDA PRECES

(Parte III)

CAPÍTULO 1.º
— PROGRAMA 3

Actualmente no mercado, existem centenas de programas monitores quer em revistas de várias nacionalidades, quer em cassetes gravadas; alguns deles bem sofisticados, outros mais simples, mas quase todos programados em linguagem máquina.

Ao ser criado no ESPAÇO SPECTRUM um capítulo destinado a programas monitores, pretendeu-se dar ao nosso leitor alguns auxiliares eficazes para a elaboração dos seus programas, sem contudo esquecer o problema capital que consiste na falta de conhecimentos técnicos da generalidade dos utilizadores portugueses. Há portanto necessidade que cada programa monitor não seja somente uma ferramenta de trabalho mas

também uma fonte de conhecimentos, possuindo uma estrutura de programação acessível que possa mais tarde ser alterada ao gosto de cada um. Para tal tem sido seleccionados programas cuja estrutura seja o BASIC, linguagem que ainda hoje é considerada como a mais próxima ao homem.

Este programa não foge a essa regra, apesar de possuir uma pequena rotina máquina. A parte do programa que se destina ao desenho e pintura do ecran está toda programada em BASIC. A rotina em C. M. comporta apenas as instruções necessárias para elaborar letras de formato (n) com tamanho e posição à escolha do utilizador. Contém esta algumas instruções algo complexas, havendo por isso necessidade de adiar a descrição do seu funcionamento para mais tarde.

O programa possui todas as instruções necessárias para a execução do trabalho a que se propõe, mas mais adiante irá sofrer pequenas modificações, e algumas das rotinas em BASIC irão passar a código máquina o que o tornará um programa muito rápido e completo.

POR QUESTÕES DE ESPAÇO, A PARTIR DO 2.º CAPÍTULO OS TEXTOS INTRODUÇÃO À LINGUAGEM MÁQUINA E ESPAÇO SPECTRUM PASSARÃO A FORMAR UM ÚNICO ARTIGO — INTRODUÇÃO À LINGUAGEM MÁQUINA ZX81 E SPECTRUM.

Noção teórica sobre o ecran do SPECTRUM em alta definição (ou resolução)

Na sua execução normalizada o Spectrum divide o ecran em duas partes distintas:

- Ecran Superior (composto de 22 linhas de 32 colunas)
- Ecran Inferior (composto de 2 linhas de 32 colunas)

A intercepção de duas linhas com duas colunas define o formato do espaço reservado aos caracteres, ou seja, em 22 linhas com 32 colunas existe um espaço que pode ser preenchido por 704 caracteres. Neste computador, o espaço preenchido por um caracter está subdividido em 64 fracções iguais, ou seja 8×8 espaços iguais. Cada um destes espaços é chamado um "pixel".

Pelo manual do Spectrum já sabemos também que é possível, com a instrução PLOT, fazer aparecer no ecran um único pixel. Claro que, como veremos, há outros processos para executar o mesmo trabalho. Também pelo manual sabemos que o Spectrum pode imprimir 256 pixels na horizontal por 176 na vertical e que essa malha está dentro dos limites definidos por alta resolução.

Para reter na memória cada um desses pixels poder-se-á pensar, à primeira vista, que são necessários $256 \times 176 = 45056$ endereços diferentes. Na realidade, apenas são precisos 6144.

Vamos ver porquê? Claro que muitos já conhecem a razão. Mas vamos esclarecer os outros.

Um endereço no Spectrum armazena um byte (8 bits) e por isso pode albergar qualquer número compreendido entre 0 e 255.

Esses 8 bits podem, como todos sabemos, tomar valores que representam as 256 combinações binárias possíveis. O mesmo é dizer que as combinações possíveis de 8 pixels podem ser armazenadas num único endereço. Então o espaço necessário na memória será

$$(256/8) = 32$$

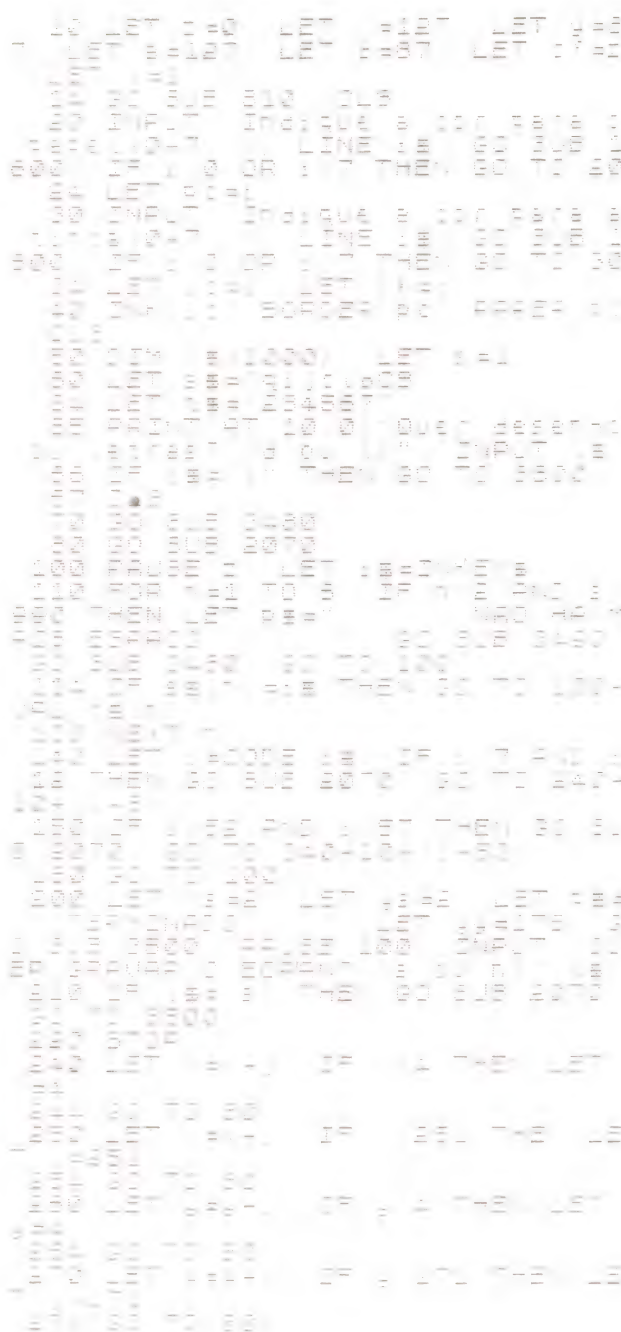
32 grupos \times 16 pixels verticais = 512 localizações para o ecran inferior

$$5632 + 512 = 6144 \text{ localizações}$$

No Spectrum, o ficheiro de projecção é guardado na RAM nos endereços (16384 a 22528) e nos endereços (22529 a 23296), espaço reservado aos atributos. Nestes não vamos ainda falar.

O programa imprime pixels no ecran com a cor da tinta pretendida, desenha círculos, quadrados, rectângulos, linhas, etc.

Quando o ecran estiver a seu gosto, pode gravá-lo em (bytes SCREEN\$) numa cassette, bastando para isso premir a tecla adequada.



$$(256/8) = 32 \text{ localizações}$$

32 grupos \times 176 pixels verticais = 5632 localizações para o ecran superior

SIMULAÇÃO DE CIRCUITOS LÓGICOS

SPECTRUM/ZX81/TS 1000

Autor: MALCOM FARNSWORTH
Adapt. de Alexandre Sousa

Parte I

PROGRAMA: SIMULAÇÃO DE CIRCUITOS LÓGICOS

DESCRIPTIVO: Este programa é interessante para todos os que gostam de Electrónica Digital, de Lógica ou de Matemática (conhecendo os dispositivos usados na Electrónica...)

Sob o ponto de vista educacional, pensamos que o programa pode ser melhorado e articulado, de molde a incluir testes para alunos ou substituir situações reais (simulação), de modo a observar o tipo de resposta a determinadas combinações lógicas dos dispositivos:



AND—OR—NAND—NOR—INVERTER—OR EX—FLIP/FLOP

ESTE PROGRAMA FOI ESCRITO PARA O ZX 81 MAS É FACILMENTE ADAPTADO AO SPECTRUM

NOTA: As linhas 7510 a 7750 — são exemplificativas da possibilidade de desenhar o próprio esquema; no entanto, sob o ponto de vista prático, parece-nos de difícil generalização, pelo que pode ser omitida a linha 4030 e a rotina apontada (7510... 7750). As linhas 5130, 6100, 7130 podem ser substituídas por uma rotina do tipo:

```
8000 input y$
8010 cls
8020 goto 4010
```

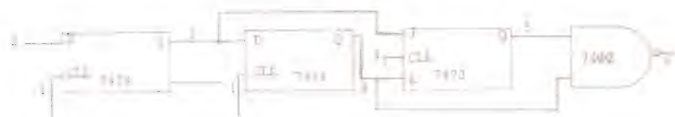
impedindo, desta forma, que o programa fique com STOP.

UTILIZAÇÃO:



Cada dispositivo lógico (AND — OR — NAND — FLIP/FLOP etc) deve ser transferido para o interior do programa.

Inicialmente, e a partir do esquema existente, vamos expor o exemplo tratado neste programa.



NUMERAMOS TODOS OS NÓS DOS DISPOSITIVOS — ENTRADA OU SAÍDA — SEQUENCIALMENTE: NÃO ESQUECENDO QUE OS NÓS LIGADOS ENTRE SI APENAS POSSUEM UM NÚMERO.

NESTE EXEMPLO TEREMOS UM NÚMERO DE NÓS =6.

Para o nosso exemplo, iremos considerar o número de impulsos de clock = 12.

No programa, a variável N corresponde a cada NÓ. O elemento mínimo de contagem do tempo, é de 1/2 período de cada impulso do "clock".



A variável correspondente aos impulsos é T.

O "array" denominado A (N,T) irá guardar os valores da temporização de cada nó (todos os nós).

Tomemos como exemplo o dispositivo NAND que está na última posição do esquema.



Na tabela da verdade correspondente, as entradas estão classificadas como 'a' e 'b' e a saída é denominada 'c'. A saída é sempre '1', excepto quando as entradas são a = 1 e b = 1. Apenas quando os nós 4 e 5 são '1', a saída do nó 'b' é zero.

Cada dispositivo lógico (AND:NAND:OR:NOR:FLIP FLOP, etc.) pertencente ao circuito ou esquema deve ser transferido para o interior do programa.

No caso tratado — 2 circuitos Flip Flop tipo D e um tipo J—K e um circuito NAND, tivemos de transferir as condições lógicas dos circuitos, para o programa:

- 1 FFlop D ; Linha 2020; linha 2030
- 2 FFlop D ; Linha 2040; linha 2050
- 3 FFlop J—K; Linha 2060; linha 2070; linha 2080
- NAND: linha 2090; linha 2100

Note que algumas das instruções derivadas de cada circuito lógico, conforme pode ver na Figura, podem ser ignoradas, dependendo da configuração do circuito. No presente caso, ambos os bi-estáveis devem ser inicialmente colocados a zero (RESET). A instrução correspondente à saída Q (negada) não é incluída no programa devido a não ser usada.

Das 5 instruções da figura flip-flop J—K serão entretanto reduzidas a 3 linhas (2060 a 2080).

Desde que os Flip-Flop sejam dependentes do clock (tempo), requerem mais atenção, como p. ex. o tipo D, caracterizado pela entrada do clock 'a'; entrada de dados 'b'; pre-set 'e'; clear 'f'.

FORMA DE 0785...

N T 123456789012



ASSERNO DO CIRCUITO

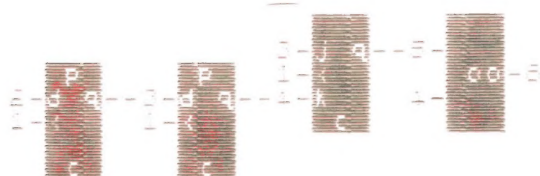
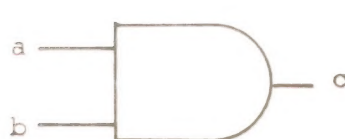


TABELA DE VERDADE
IMP. - NUM. de NO

	1	2	3	4	5	6
1	0	0	0	0	0	1
2	1	0	0	0	0	0
3	0	1	0	0	0	0
4	0	0	1	0	0	0
5	0	0	0	1	0	0
6	1	1	1	1	1	1
7	0	0	0	0	0	0
8	1	0	0	0	0	0
9	0	1	0	0	0	0
10	0	0	1	0	0	0
11	0	0	0	1	0	0
12	1	1	1	1	1	1

0 000101
1 0 100001
1 1 000001
1 1 1 000001
1 1 1 000001

A N D

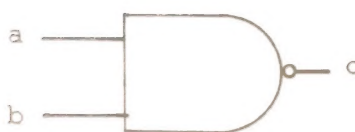


a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

1) LET A(C,T)=0

2) IF A(A,T)=1 AND A(B,T)=1 THEN
LET A(C,T)=1

N A N D



a	b	c
0	0	1
0	1	1
1	0	1
1	1	0

1) LET A(C,T)=1

2) IF A(A,T)=1 AND A(B,T)=1 THEN
LET A(C,T)=0

(Cont. no próximo número)

MICRODRIVE

SPECTRUM

INTERFACE

(Continuação)

O interface é o dispositivo que permite ao SPECTRUM comunicar com os periféricos de entrada/saída, do tipo impressora ou Microdrive.

O interface que ficará ligado ao SPECTRUM será do tipo RS-232 e será denominado INTERFACE 1.

É de notar que já foi lançado em Inglaterra o Interface 2 que será usado para ligação de JOYSTICK's e programas em ROM.

O INTERFACE RS-232 estabelece a comunicação entre dispositivos diferentes e, no caso da SINCLAIR, tem uma ficha de 9 pinos, dos quais dois são para recepção de sinais e outros dois para remeter dados.

Os dados são transferidos entre o SPECTRUM e o MICRODRIVE no modo SÉRIE, ou seja, cada *bit* segue sequencialmente através da linha de dados; por isso, teremos um outro dado em jogo, que é a velocidade de transmissão ou BAUD RATE, com valores entre 50 e 19200 bits por segundo.

DESCRIPTIVO DO MICRODRIVE

Trata-se de uma caixa c/ 90 x 85 x 40 mm com uma abertura na parte da frente, para introdução do cartucho.

Possui uma ficha do lado direito, onde é ligado um cabo flexível, que estabelece a comunicação com o INTERFACE 1.

No seu interior existem dois circuitos impressos. O que está situado na posição inferior, é onde se estabelecem as duas ligações para o exterior.

O circuito central (vertical), possui uma memória ROM e a cabeça de reprodução/gravação, a qual possui as duas funções: GRAVAR/LIMPAR.

Quando o microdrive está em funcionamento, um indicador (LED) fica luminoso.

- NÃO DEVE NUNCA, REMOVER O CARTUCHO, QUANDO O "LED" ESTÁ LUMINOSO.
- RETIRE O CARTUCHO QUANDO LIGAR OU DESLIGAR A ALIMENTAÇÃO.

O cartucho possui as dimensões de 45 x 35 x 7 mm e possui protecção contra gravação. Essa protecção deve ser retirada, quando pretendemos efectuar qualquer gravação.

Um cartucho contém uma só fita, de 5 metros de comprimento e com a largura de 1,5 mm. Trata-se de uma peça de fita girando em contínuo.

USAR O MICRODRIVE

Podemos usar o *Microdrive* após ter sido feita a ligação do *Interface 1* ao SPECTRUM (a alimentação será a do Spectrum).

Para utilização do dispositivo, necessitamos de um cartucho ou *Floppy Tape*. Voltamos a referir que não se trata de uma *diskette* vulgar, mas sim de uma fita, mecanicamente muito rápida.

A primeira operação será a de FORMATAR o cartucho que está virgem. A formatação dura cerca de 30 segundos, durante os quais o

rectângulo do ecran (conhecido como *Border*) ficará em acção de *flash*. No final teremos uma mensagem de OK.

A acção de formatar (ou inicializar) é fundamentalmente um conjunto de operações de definição de áreas, onde o computador pode ler ou escrever e também demarcar as áreas que não podem ser usadas. A repetição desta operação leva à destruição de todo o conteúdo da fita, e só é efectuada quando queremos usar uma fita cujos dados já não têm interesse para nós.

A instrução que permite a inicialização da fita tem o nome de **FORMAT** e será usada com o *microdrive*, com o *interface RS-232* e com a rede local (*Network*).

FORMAT — "m"; 1; "TESTE"

FORMAT — instrução

"m"; 1 — *microdrive* 1

"Teste" — nome que pode ter até 10 caracteres

OUTRAS INSTRUÇÕES:

- **CAT 1** — será a instrução usada para obter a lista de todos os nomes de programas ou conjuntos de dados que gravou na fita.
- **SAVE * "m"; 1; "TESTE"** — destina-se a gravar um programa na fita que está no *Microdrive* 1. O asterisco indica ao computador que o **SAVE** se destina ao *microdrive* e não ao gravador. "m"; 1 significa *microdrive* 1 e "TESTE" será o nome do programa. Um cartucho pode estar protegido para que não se possa escrever na fita, se lhe for retirada uma pequena peça plástica.
- **VERIFY * "m"; 1; "TESTE"** — verifica se o programa foi gravado anteriormente.
- **LOAD * "m"; 1; "TESTE"** — executa a reprodução de um programa gravado na fita.
- **MERGE** — permite juntar 2 programas num só. Pode ser usada desde que nenhum dos programas tenha sido gravado com **SAVE** **LINE** número.

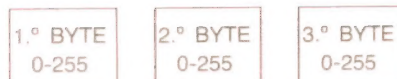
Uma das inovações significativas é poder-se preparar um programa com **AUTO-RUN**, ou seja, ligar o computador e arrancar com um determinado programa.

IMPORTANTE! — O comando **RUN** do **SPECTRUM** deixa de funcionar e passa a escrever-se **R... U... N**, letra a letra.

O programa será gravado sob a forma:
SAVE * "m"; 1; "RUN" LINE 100
 (ou outro número)

Entrando a palavra **RUN**, a partir do teclado, o programa será reproduzido corrido a partir da linha 100.

FLOPPY TAPE (Cartucho de Fita)



No desenho acima pretendemos representar a forma como ficam gravados os *bits* e os *bytes* (**BYTE** = 8 **BITS**) ao longo de uma fita. Todos os dados são gravados na fita, em duas pistas (*track*), e os *bits* (um *bit* é a unidade de informação — 0 ou 1) são dispostos alternadamente em cada uma das pistas.

Em princípio não nos devemos preocupar se o facto de o *Microdrive* ter duas cabeças o torna mais lento ou sobre o comprimento que ocupa um *bit* ao longo de uma fita.

O bloco de **DADOS** (*data*) inicia-se com uma frente de 12 bytes, dos quais 10 são zeros e dois são "255" (255 em hexadecimal são FF, e em binário são **////////**) — isto destina-se a um perfeito reconhecimento do início de um bloco de dados.

MICRODRIVES

RESPOSTAS A ALGUMAS QUESTÕES

PODEMOS USAR OS PROGRAMAS TRADICIONAIS?

Em primeiro, a maioria dos programas já existentes funcionam; apenas nos casos em que esses programas fazem uso do "interrupt control", os programas não funcionarão!

De qualquer modo, se o programa já existir quando obteve o *microdrive*, não tente usar o *microdrive* ou a rede externa, mas use o programa exactamente como se o *microdrive* não existisse.

QUAL O NÚMERO MÁXIMO DE FILES QUE PODEMOS GRAVAR NUM CARTUCHO?

Cada file ocupa um mínimo de 512 Bytes ou seja um "buffer" ou tampão.

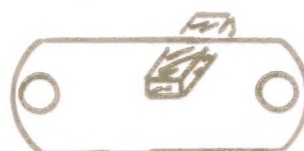
O catálogo ou directório das files mostra um número máximo de 50 nomes de files, embora o excesso, permaneça na fita e possa ser chamado em qualquer altura.

QUAL O TEMPO MÍNIMO PARA GRAVAR UMA FILE?

Será preferível falar em tempo médio, que no caso de Files de curta capacidade, será cerca de 10 segundos.

QUAL O TEMPO MÉDIO PARA ENCONTRAR UMA FILE ARQUIVADA NUM CARTUCHO?

O sistema usado pelo *microdrive*, é descrito desta forma:



Não existe enrolamento da fita mas sim uma passagem em contínuo.

Se a ficha (file) que pretendemos ler se encontra para além da cabeça de leitura, teremos de esperar que a zona da fita se situe junto da cabeça de leitura. Deste modo teremos entre 1 a 7 segundos para encontrar uma ficha!



CLUBE Z₈₀

INSCRIÇÃO COMO ASSOCIADO

O **CLUBE Z₈₀** está aberto a todos os utilizadores de microcomputadores.

A intenção de associar os entusiastas das micro-máquinas, é exclusivamente a de permitir:

- 1 — PUBLICAÇÃO DE UM JORNAL MENSAL, onde sejam publicados programas de uso geral ou específico como no caso da educação.
- 2 — PROMOVER TROCAS DE PROGRAMAS, e trocas de experiências; tanto no caso do Software (programação), como no caso do Hardware (electrónica).
- 3 — PROMOVER DESCONTOS NA AQUISIÇÃO DE PROGRAMAS.
- 4 — LANÇAR CURSOS DE PROGRAMAÇÃO EM BASIC — PASCAL OU OUTRAS LINGUAGENS E DIVULGAR O USO DE LINGUAGEM MÁQUINA.

NOME

IDADE COMPUTADOR TIPO

PROFISSÃO

ENDEREÇO

TELEF.

ASSINATURA ANUAL — Esc. 1 500\$00 ☐

ASSINATURA SEMESTRAL — Esc. 750\$00 ☐

CHEQUE OU VALE DO CORREIO

N.º

BANCO

DATA/...../.....

JÁ SÓCIO ☐

NOVO SÓCIO ☐ → A partir do mês de (inclusive)